



W01

INTRODUCTION TO **JAVA**

Thakshila Dasun
BSc. Eng in Mechatronics Eng
CIMA (UK)

The Evolution Of Computer Programming Languages

```

0D 54 60 64 00 00 00 06 00
72 6B 00 00 00 61 00 F0 0A
00 41 F7 00 B0 00 09 00 C0
EA 22 01 00 00 00 00 00 00
00 00 00 FF 61 03 06 0A 1B
6A 6F 2E 00 30 42 40 2C 80
30 42 00 04 70 41 5E 2C 00
80 40 00 00 FF 2F 00 4E 54
88 00 00 00 C8 04 00 FF 7F
00 00 00 00 00 00 02 00 00
03 07 6E 2D 70 69 41 4E 4F

```



Hex

NAME	DATE	TIME
SCHOOL	DAY	1
STATE	CITY	9861.0
POSTOFFICE	STREET	P.O.

GRADE	PERCENT	SEX
AGE	AVERAGE	HEIGHT
WEIGHT	HAIR	
COMPLEXION	ARTS	ENTREPRENEURSHIP

RELIGION	DEGREE	SKILL
HOBBIES	INTERESTS	FOCUS
ACHIEVEMENTS	GOALS	PROGRESS
REFERENCES	ADVICE	FEEDBACK



Assembler

```
#include <stdio.h>
#include <io.h>
#include <dos.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

main()
{
    char ch,*test;
    int bit;
    int data;
    FILE *f;
    struct
    {
        char
```



C



Fortran

```

classset: greguana do
class: pteropoda {
public:
    int x;
    void draw() { gpguires(x,y,WHITE);
}

class: carapoda: public: pteropoda {
public:
    void swim();
    void draw() { class(x,y,z,redraw);
}

main()
{
    int screen;
    desktop a w;
    menu = 0;
    gpguires(0,0,0,0);
    gpguires(0,0,0,0);
    while(1)
    {
        if
    }
}

```



C++

[illegible]

Java

```

package org.apache.commons.logging.impl.jdk14;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.apache.commons.logging.LogNotFoundException;
import org.apache.commons.logging.LogManager;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.PrintStream;
import java.util.Enumeration;
import java.util.HashMap;
import java.util.Map;
import java.util.Properties;
import java.util.concurrent.ConcurrentHashMap;
import java.util.concurrent.ConcurrentMap;
import java.util.concurrent.atomic.AtomicInteger;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.apache.commons.logging.LogNotFoundException;
import org.apache.commons.logging.LogManager;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.PrintStream;
import java.util.Enumeration;
import java.util.HashMap;
import java.util.Map;
import java.util.Properties;
import java.util.concurrent.ConcurrentHashMap;
import java.util.concurrent.ConcurrentMap;
import java.util.concurrent.atomic.AtomicInteger;

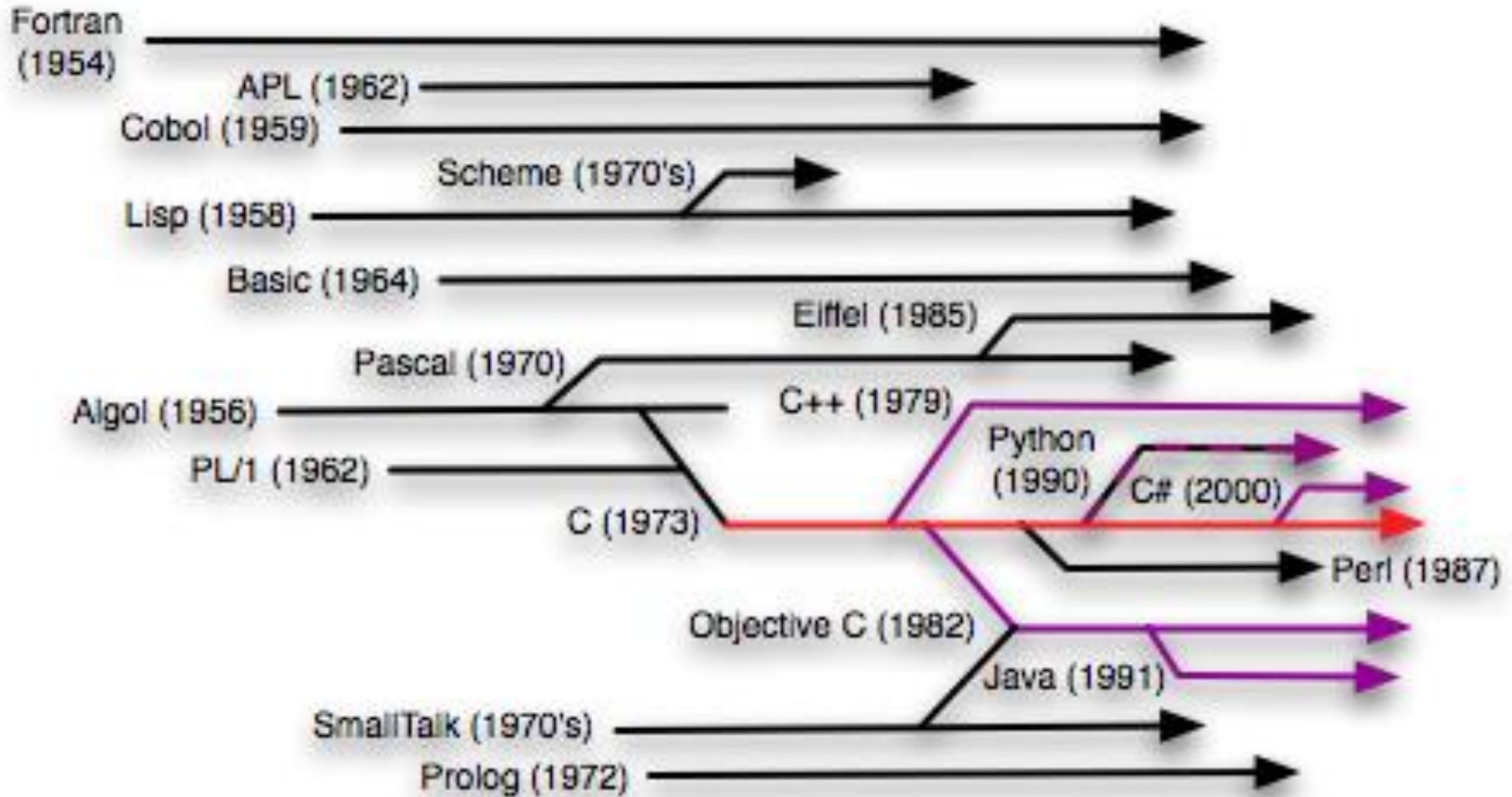
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

```



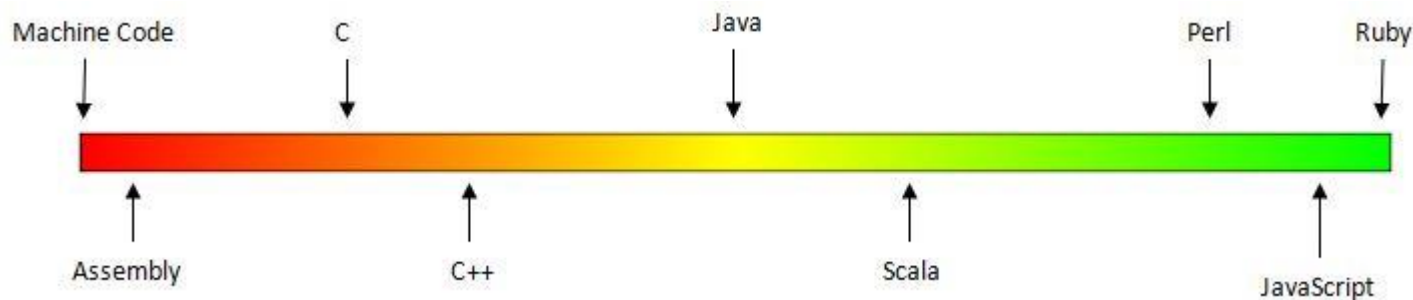
Ruby

NOW This



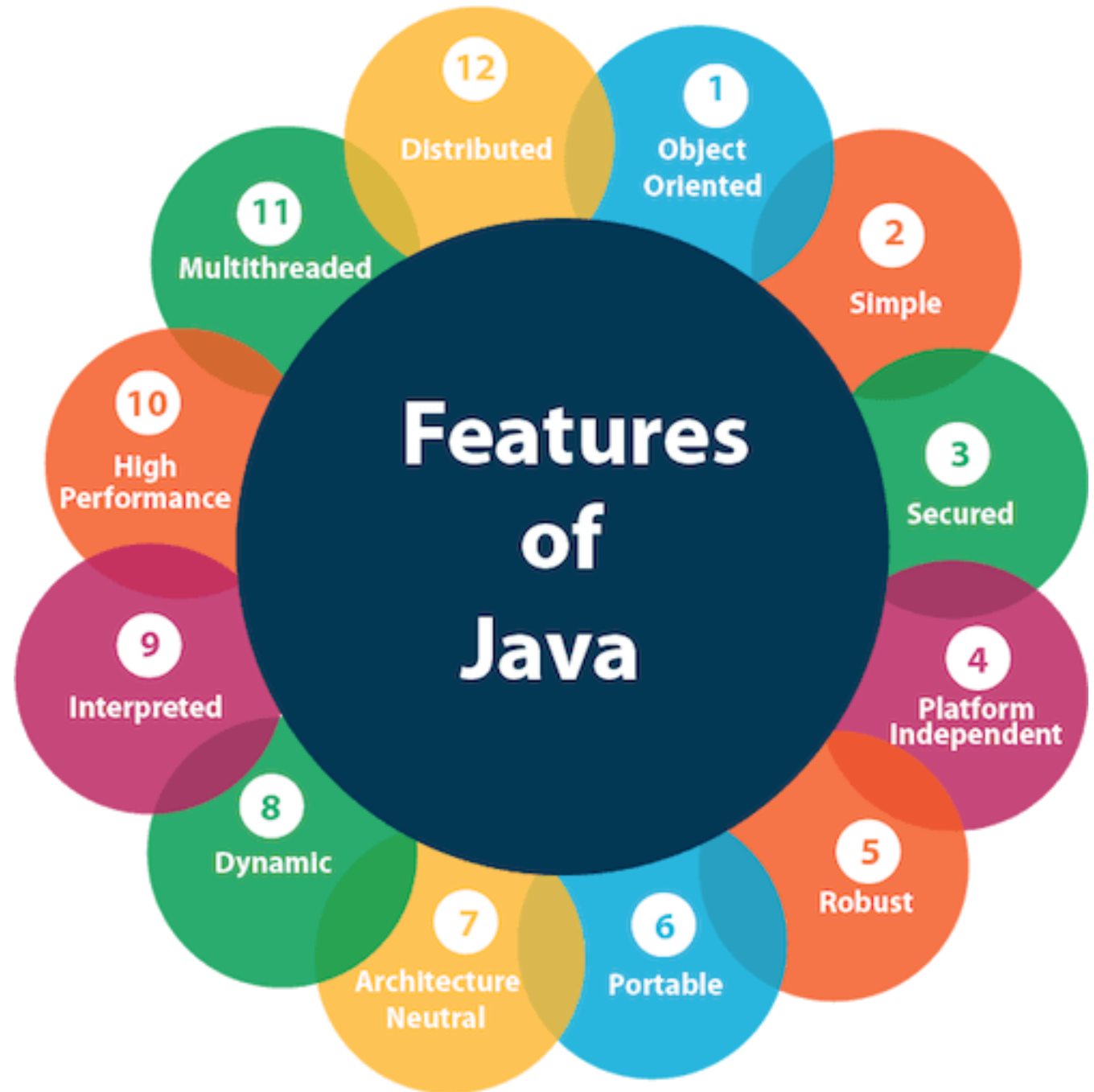
INTRODUCTION (1) *“Write Once, Run Anywhere”*

- Java is a high-level programming language.
- Originally developed by Sun Microsystems and released in 1995.
- Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX
- JAVA is a **Object-Oriented Programming Language**.



Note: A programming language is said to use static typing when type checking is performed during compile-time as opposed to run-time.

Why Java?



INTRODUCTION (2)

- **Object Oriented** – In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
- **Platform Independent** – Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.
- **Simple** – Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.

INTRODUCTION (3)

- **Secure** – With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- **Architecture-neutral** – Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
- **Portable** – Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

INTRODUCTION(4)

- **Robust** – Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.
- **Multithreaded** – With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.
- **Interpreted** – Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.

INTRODUCTION(5)

- **High Performance** – With the use of Just-In-Time compilers, Java enables high performance.
- **Distributed** – Java is designed for the distributed environment of the internet.
- **Dynamic** – Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

Why JAVA and what is the usage? (1)

- Create **Web Applications**
- Build **Applications and Platforms** for a number of devices, including computers, laptops, gaming consoles, Blu-ray players, car navigation systems, medical monitoring devices, parking meters, lottery terminals and smartphones.
- It is also a key language for **Networking**, particularly for data centers that store and transfer Web-based data.
- **Google Recommends for Android App Development**
- **Loads of Frameworks, Libraries, IDEs and Development Tools**

Why JAVA and what is the usage? (2)

- **Simplify Development of Real-Time Software**
- **Supports Internet of Things**